JD Jamieson

CS273

Final Specifications

**Requirements:**

My final project is a hospital emergency room simulator. When the program is initiated, the user will be asked to input how many doctors are working in the ER, how many nurses are in the ER, and what the average hourly rate of new patient arrivals is. After storing these values, the simulator will start, simulating a week in this emergency room. When a patient arrives, they will be given a priority value depending on the severity of illness. Each severity, ranging from 1 to 20, occurs at a fixed probability. The patient will also be given a start value depending on the clock, to designate when they arrived at the hospital. The simulator will then check to see if there any caregivers available to treat the patient. If the severity value of the patient is between 1 and 10, nurses will be searched for first, to keep the doctors free for higher value patients or large numbers of patients. If no nurse is available, then doctors will be searched for. Patients with severity values from 11 to 20 will bypass the nurse search and go to the doctor search. If no caregivers are immediately available, the patient will be put in a priority queue, where high severity patients will be given first priority over low severity. Once assigned a caregiver, the patient will be treated. Nurses treat patients on average for 1 to 10 minutes, while doctors take 1 to 20 minutes. Once treated, the patient’s visit time will be calculated by taking the clock value at dismissal and subtracting the stored start time for arrival. The visit will be stored in the hospitals records, showing how many times a patient visited the ER in the week and the severity of each visit. Once the simulation has run through a week (10,080 minutes), the average visit time will be calculated. Also, a menu will be displayed allowing the user to see who all visited the ER or get the records stored for a given name.

**Use Cases:**

* **Case 1: Initializing the simulation**
  + User opens up program
    - Program prompts user for number of doctors in ER
  + User inputs number of doctors
    - Program prompts for number of nurses in ER
  + User inputs number of nurses
    - Program prompts for average patient arrival rate
  + User inputs arrival rate
    - Program runs simulation
* **Case 2: Displaying the records/Individual search**
  + User issues command to display patient records
    - Program outputs names of patients who visited ER
  + User issues command to search by name
    - Program prompts for name
  + User inputs name
    - Program searches records for name. Outputs all records matching given name. If name is not in records, program displays message saying name was not found.

**UML:**

Doctor

Priomax=20

treatTime=20

busy=false

Priomax=10

treatTime=10

busy=false

Caregiver

Nurse

-Int PrioMax

-Int treatTime

-Bool busy

+treatPatient()

+checkBusy()

+changeBusy()

Hospital

-String name

-Priority\_queue<Patient> waiting\_room

-Vector<Caregiver\*> staff

-Int clock

-Int arrival\_rate

-Multimap<string, int> records

-Int num\_served

-Int avg\_time

+void triage(Patient)

+void setRate(int)

+void setStaff(int,int)

+void admitPatient()

+void checkArrivals(int)

+void recordVisit(Patient, int)

+int calc\_time

+void display\_records()

+void search\_records(string)

Patient

-string name

-int arrival\_time

-int severity

-int treatment\_time

+int getArrival();

+string getName()

+int getSeverity()

+int getTreatmentTime()

+bool operator<(Patient)

+void setSeverity(int)

+void setTime(int)

+void setTreatment(int)

**Pseudo-code:**

Void TreatPatient(Patient){

If(Caregiver has a patient){

Check if patient is done with treatment

If yes{

Add visit to records

Dismiss patient

Change flag to free

}

If (Caregiver isn’t busy){

Take new patient

Change flag to busy

}

}

Void DisplayRecords(){

Multimap<string, int> iterator mit=map.begin()

While (iterator!=map.end())

Display name on Record

}

Void searchRecords(string name){

Set up iterator (same as in DisplayRecords)

While (iterator != map.end())

{

If (name on file matches parameter name)

Display severity

Else

Skip to next file

}

}